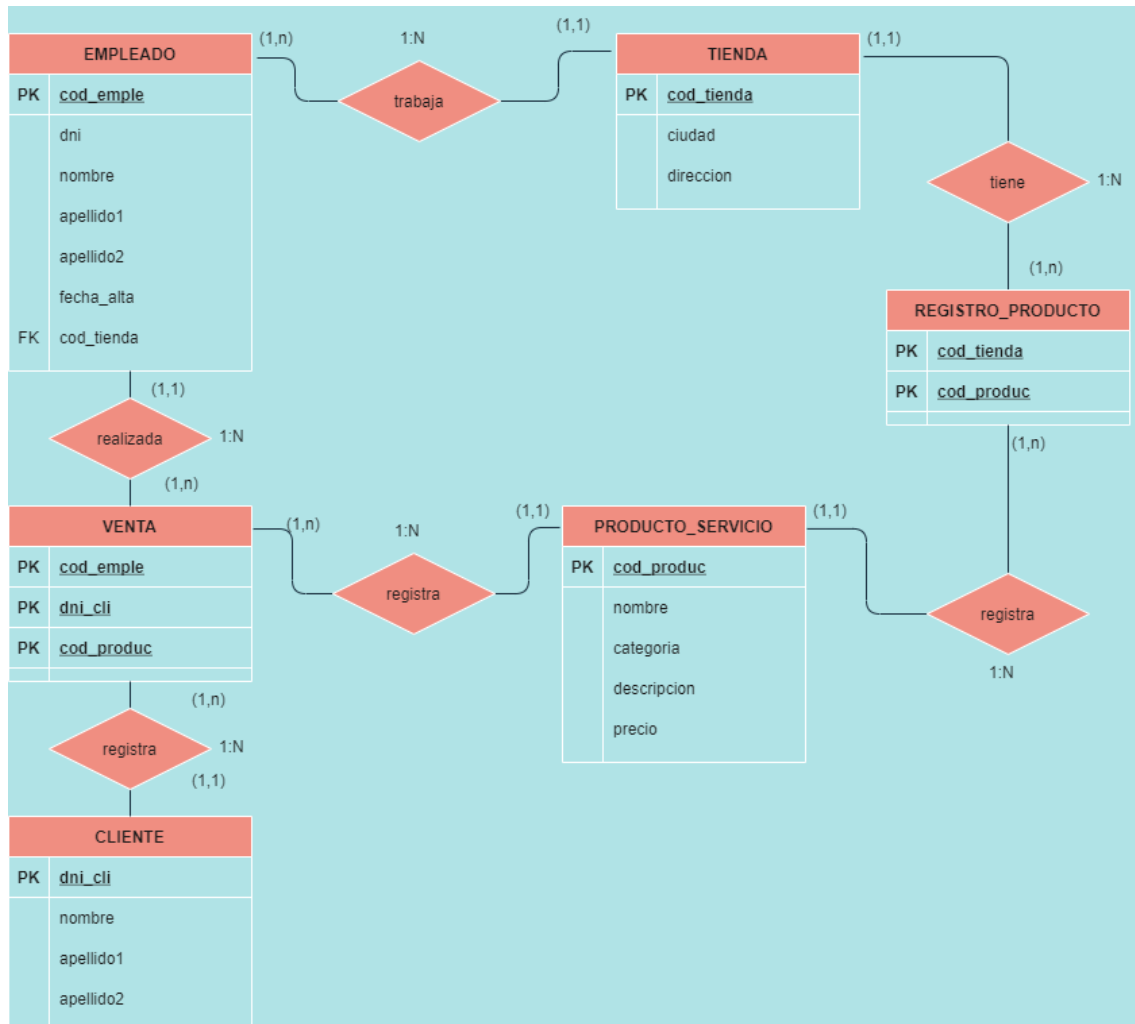


Proyecto PL/SQL

Modelo Entidad-Relación.



Modelo Relacional

Tablas	Detalles
TIENDAS	cod_tienda(PK) ,ciudad,direccion.
EMPLEADOS	cod_emple(PK) ,dni,nombre,apellido1,apellido2,fecha_alta, cod_tienda(FK) . cod_tienda: foreign Key de tienda.
CLIENTES	dni_cli(PK) ,nombre,apellido1,apellido2.
PRODUCTOS_SERVICIOS	cod_produc(PK) ,nombre,categoría,descripcion,precio.
VENTAS	cod_emple(PK),dni_cli(PK),cod_produc(PK) cod_tienda: foreign Key de Tiendas. cod_produc: foreign Key de Productos_Servicios. dni_cli: foreign Key de Clientes.
REGISTROS_PRODUCTOS	cod_tienda(PK), cod_produc(PK) cod_tienda: foreign Key de Tiendas. cod_produc: foreign Key de Productos_Servicios.

Descripción de la Base de Datos.

Base de datos para gestionar tiendas de mascotas junto a sus productos y ventas a clientes.

Consta de las siguientes Tablas:

- 1 clientes
- 2 empleados
- 3 productos_servicios
- 4 venta
- 5 tienda
- 6 registro_producto

La tabla clientes tiene 4 columnas:

- DNI_CLI -- Identificador del cliente siendo este su DNI
- NOMBRE -- Nombre del cliente (no puede ser nulo)
- APELLIDO1 -- Apellido paterno del cliente pudiendo ser un valor nulo
- APELLIDO2 -- Apellido materno del cliente pudiendo ser un valor nulo

Consta de las siguientes constraints:

-CONSTRAINT CLI_DNI_PK PRIMARY KEY, -- Indica que la primary key de la tabla es el atributo DNI_CLI

La tabla empleados tiene 7 columnas:

- COD_EMPLE -- Identificador de la tabla para cada empleado siendo un código de hasta 4 dígitos

DNI -- identificador personal de cada empleado.

NOMBRE -- nombre del empleado (no puede ser nulo)

APELLIDO1 -- Apellido paterno del empleado pudiendo ser un valor nulo

APELLIDO2 -- Apellido materno del empleado pudiendo ser un valor nulo

FECHA_ALTA -- Fecha de alta del empleado (no puede ser nulo)

COD_TIENDA -- Código de la tienda en la que trabaja el empleado.

Consta de las siguientes constraints:

-CONSTRAINT EMP_COE_PK PRIMARY KEY, -- Indica que la primary key de la tabla es el atributo COD_EMPLE

-CONSTRAINT EMP_DNI_UQ UNIQUE , -- Comprueba que el DNI sea único para cada empleado.

-CONSTRAINT EMP_COT_FK REFERENCES TIENDA(COD_TIENDA) --Referenciamos el atributo COD_TIENDA a su respectiva tabla

La tabla PRODUCTOS_SERVICIOS tiene 5 columnas:

COD_PRODUC -- Identificador de cada producto siendo un código de hasta 4 dígitos

NOMBRE -- Nombre del producto (no puede ser nulo)

Categoria -- Categoría del producto (no puede ser nulo)

DESCRIPCION -- Descripción del producto la cual es posible que sea un valor nulo

PRECIO -- Precio del producto siendo posible ingresar un valor de 4 dígitos con 2 decimales, por defecto será 10.

Consta de las siguientes constraints:

-CONSTRAINT PRS_COP_PK PRIMARY KEY, -- Indica que la primary key de la tabla es el atributo COD_PRODUC

-CONSTRAINT PRS_PRE_CH CHECK(PRECIO>0) -- Constraint que comprueba el valor del precio sea mayor que 0

La tabla venta tiene 3 columnas:

COD_EMPLE -- Código del empleado que vende el producto

DNI_CLI -- Código del cliente que compra el producto

COD_PRODUC -- Código del Producto que se vende al cliente

Consta de las siguientes constraints:

-CONSTRAINT VEN_PK PRIMARY KEY(COD_EMPLE,DNI_CLI,COD_PRODUC) -- Constraint que indica que los tres valores anteriores son primary key

-CONSTRAINT VEN_COE_FK REFERENCES EMPLEADOS(COD_EMPLE) -- Referenciamos el atributo COD_EMPLE a su respectiva tabla

-CONSTRAINT VEN_DNC_FK REFERENCES CLIENTES(DNI_CLI) -- Referenciamos el atributo DNI_CLI a su respectiva tabla

-CONSTRAINT VEN_COP_FK REFERENCES PRODUCTOS_SERVICIOS(COD_PRODUC) -- Referenciamos el atributo COD_PRODUC a su respectiva tabla

La tabla tienda tiene 3 columnas:

COD_TIENDA -- Identificador de la tienda siendo un código de hasta 4 dígitos

CIUDAD -- Nombre de Localidad a la que pertenece la tienda (no puede ser nulo)

DIRECCION -- Dirección concreta de la tienda (no puede ser nulo)

Consta de las siguientes constraints:

-CONSTRAINT TIE_COT_PK PRIMARY KEY -- atributo COD_TIENDA primary key de la tabla tienda

-CONSTRAINT TIE_DIR_UQ UNIQUE(DIRECCION) -- CONSTRAINT unique para la dirección ya que no va a ver dos direcciones iguales para distintas tiendas.

La tabla registro_producto tiene 2 columnas:

COD_TIENDA -- Código de la tienda donde está el producto.

COD_PRODUC -- Código del producto que tiene una tienda.

Consta de las siguientes constraints:

-CONSTRAINT REP_COT_FK REFERENCES TIENDA(COD_TIENDA) -- Referenciamos el atributo COD_TIENDA a su respectiva tabla

-CONSTRAINT REP_COP_FK REFERENCES PRODUCTOS_SERVICIOS(COD_PRODUC) -- Referenciamos el atributo COD_PRODUC a su respectiva tabla

-CONSTRAINT REP_PK PRIMARY KEY(COD_PRODUC,COD_TIENDA) -- Indicamos que las dos columnas son la primary key de la tabla.

Requisitos funcionales de la base de datos.

Mi base de datos se centra en llevar un registro de las ventas y los productos que tenemos en la tienda junto con quien los vende.

Además, consta de 2 funciones y 3 procedimientos:

Las funciones se centran en darnos en darnos quien a sido el mejor cliente y mejor vendedor de las tiendas.

Los procedimientos dos de ellos nos ofrecen un catalogo uno de los empleados de cada tienda y el otro es un catalogo de los productos que ofrecemos filtrado por categorías, además hay un procedimiento que complementa la función de mejor cliente y nos devuelve el nombre de ese cliente.

Mi bloque anónimo nos da un formato a los procedimientos y funciones mencionados anteriormente.

Enlace a GitHub

https://github.com/ManuelRuJi/Proyecto_PL-SQL.git